Course: Software Project Management

Week 9: Risk Strategies, Monitoring & Controlling

Lecturer: Yimer Amedie (MSc.)

Addis Ababa Science and Technology University, Ethiopia

Contents

- Introduction
- Risk Strategies
- Risk Monitoring and Controlling
- Issue management

Learning Outcomes

After completing this lesson, you will be able to:

- Explain key risk response strategies used in software project management.
- Monitor and track project risks using basic risk management tools and adjust plans as needed.
- Differentiate between risks and issues, and describe the steps involved in managing project issues effectively

Introduction

- In risk management, the following are the foundational stages:
  - Risk planning and identification
  - Risk assessment/analysis
  - Risk responses strategies
  - Risk monitoring & controlling
- So, the last two stages will be discussed in this lecture.
- In addition, issue management is discussed here, a distinct but closely related concept with risk management.
  - Deals with actual problems impacting the project

Project Risk Management Process

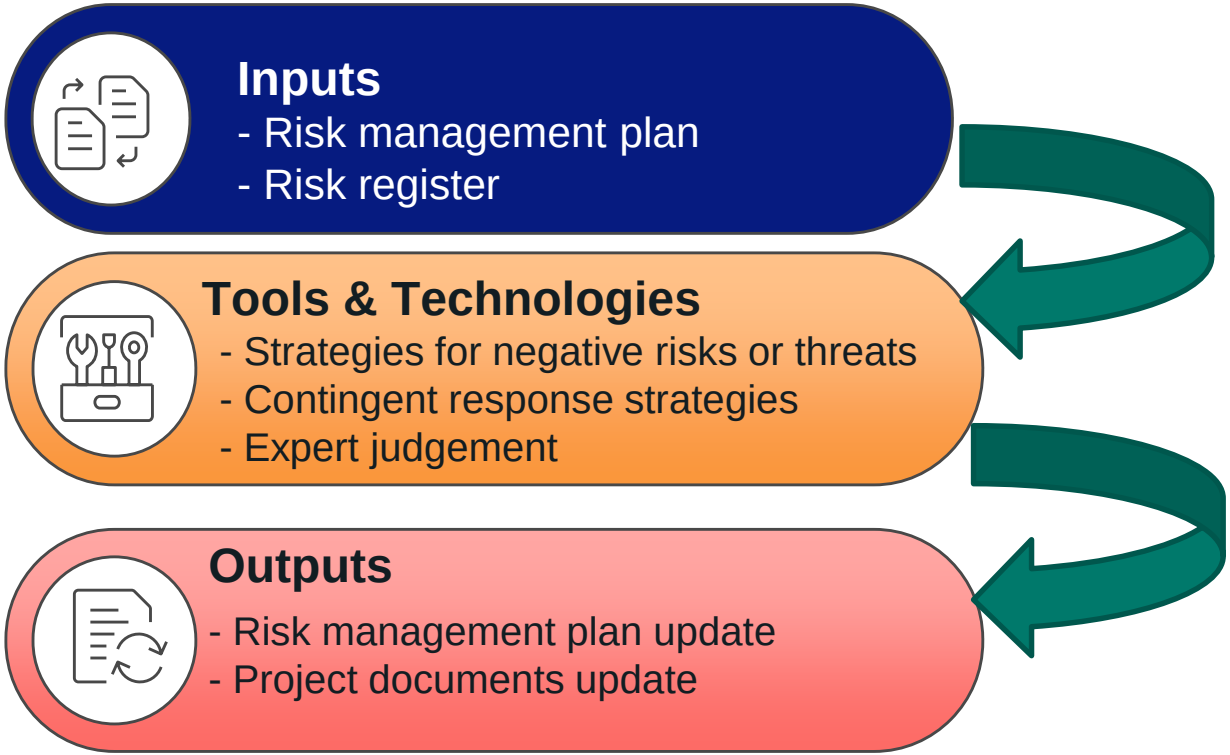
Risk Strategy/responses

- After assessing risks, it is essential to develop mitigation strategies to address them.
- It is a plan for responding the risks if occurred.
- Plan risk responses is the process of developing options and actions
  - To reduce threats to project objectives
  - To enhance opportunities

Project Risk Management Process

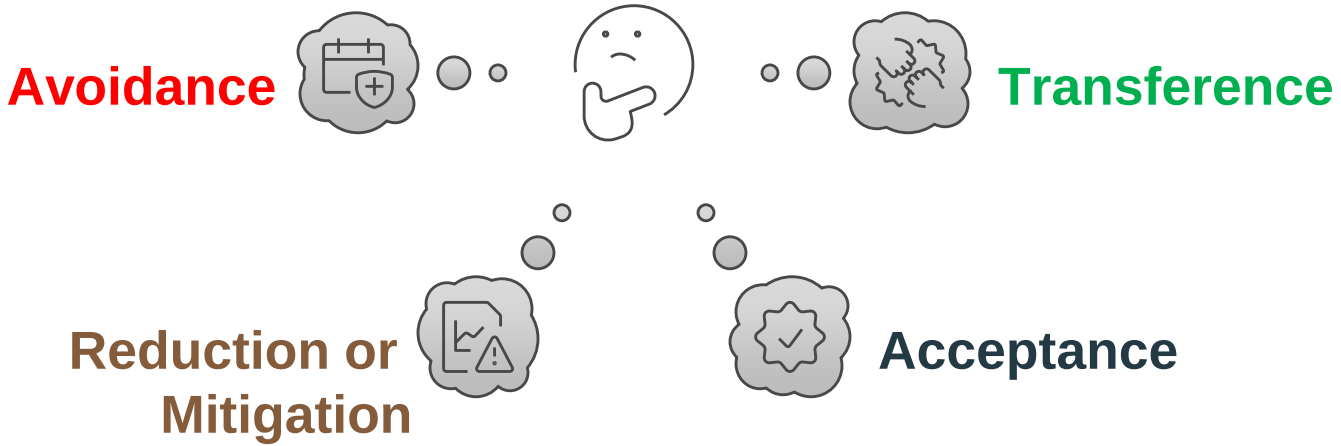
Plan Risk Response: Inputs, Tools & Technologies and Outputs

(PMI, 2013)



Project Risk Management Process

Common strategies for negative Risks (Bob Hughes, 2011)



Project Risk Management Process

Common strategies for negative risks - Avoidance

- Altering project plans to eliminate the risk entirely.
- Eliminating the risk entirely by removing its cause or choosing not to engage in a risky activity.
- It’s a proactive and preventive strategy.
- The best option when the impact of the risk is high and not worth taking.

Project Risk Management Process

Common strategies for negative risks - Avoidance

- **Example, Online Learning Management System**
  - **Risk:** The LMS fails to integrate smoothly with SIMS.
  - **Strategy:** Avoidance
  - **Action:** Engage both LMS and SIMS vendors to review integration feasibility. Align LMS database schema with the SIMS to ensure clean mapping.
  - **Owner:** Development Team
  - **Time:** Week 11

Project Risk Management Process

Common strategies for negative risks - Reduction or Mitigation

- Implementing measures to reduce the likelihood or impact of the risk.
- Unlike risk avoidance, which involves removing the risk entirely, mitigation focuses on making the risk manageable.
- It is essential because not all risks can be avoided, and trying to do so might delay the project too much or increase costs.
- Instead, mitigation plans aim to make these risks less severe.

Project Risk Management Process

Common strategies for negative risks - Reduction or Mitigation

- **Example: Online Learning Management System**
  - **Risk:** Data Loss or Corruption During Integration with SIMS
  - **Strategy:** Mitigation
  - **Action:** Implement regular data backups to prevent data loss during integration.
    - Use a staging environment for initial integrations, where data can be tested and validated before going live.
  - **Owner**: Development team. **Time**: Week 13

Project Risk Management Process

Common strategies for negative risks – Transference

- Shift the responsibility of managing a risk to another party, typically a third party such as a **vendor**, **contractor**, or **insurance provider**.
- This doesn't eliminate the risk, but it minimizes your exposure to the impact of that risk by having someone else handle it.
  - In software development projects, risk transference can be used to manage several types of risks, such as
    - **Technical issues, project delays, security breaches, or infrastructure problems.**

Project Risk Management Process

Common strategies for negative risks – Transference

- **Example: Online Learning Management System**
  - **Risk:** Delays or failure in meeting technical milestones, such as building custom integrations or features, due to limited in-house expertise.
  - **Strategy:** Transference
  - **Action:** Outsource complex development tasks to specialized software development company or hire contract developers with expertise in specific technologies (e.g., APIs, machine learning, etc.).
  - **Owner:** Vender/PM**. Time:** Week **2-3**

Project Risk Management Process

Common strategies for negative risks - Acceptance

- Acknowledging the risk and preparing to deal with its consequences if it occurs.
- Often because the impact is either low, tolerable, or manageable within the project's scope.
- Used when the cost of preventing or mitigating the risk exceeds the potential impact or when the risk is inevitable

Project Risk Management Process

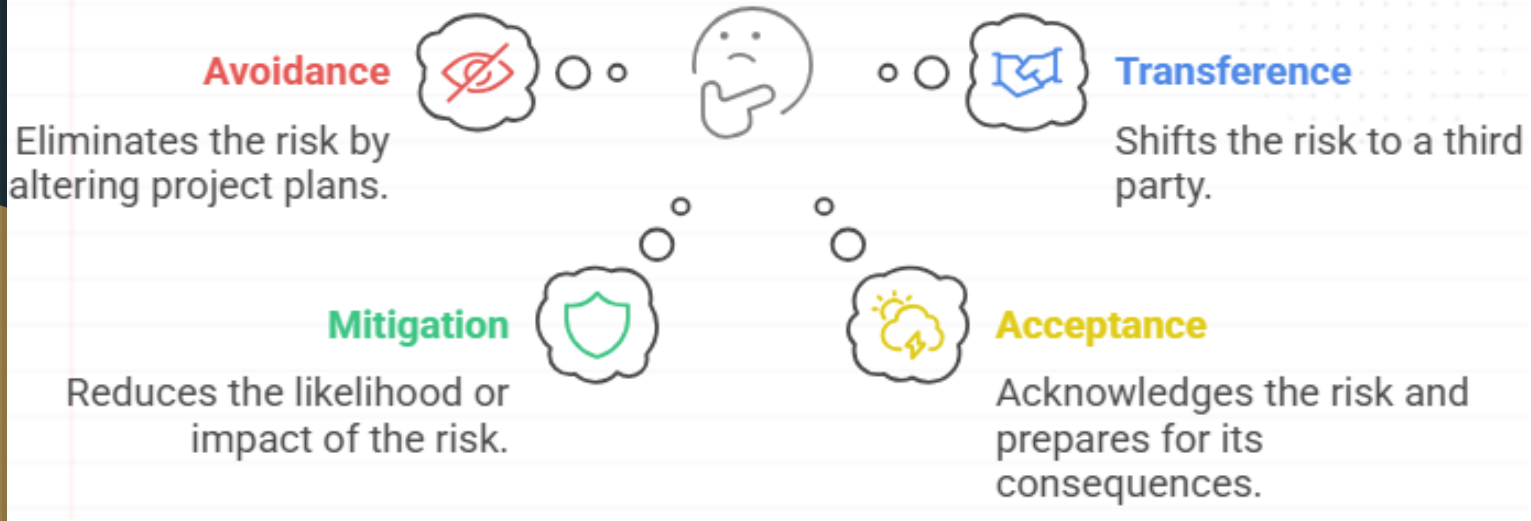
Common strategies for negative risks - Acceptance

- **Example: Online Learning Management System**
  - **Risk:** SIMS integration might not be perfect from day one, resulting in delays or data synchronization issues in the first few weeks.
  - **Strategy:** Acceptance
  - **Action:** Accept that temporary delays in syncing data might happen and keep users informed.
    - Include this in the project plan as a short-term inconvenience.
  - **Owner**: Team, PM. **Time**: Week 18

Project Risk Management Process

Risk Strategy/responses

- Which risk mitigation strategy should be implemented, when?



Project Risk Management Process

Risk Strategy/responses - Activity

- Assume you're managing a software development project for a mobile banking app. The deadline is tight, and you rely heavily on a third-party encryption library that hasn't yet been updated to work with your current tech stack.
    1. Identify the risk(s)
    2. Choose an appropriate response strategy.

Project Risk Management Process

Risk Strategy/responses – Activity - Solution

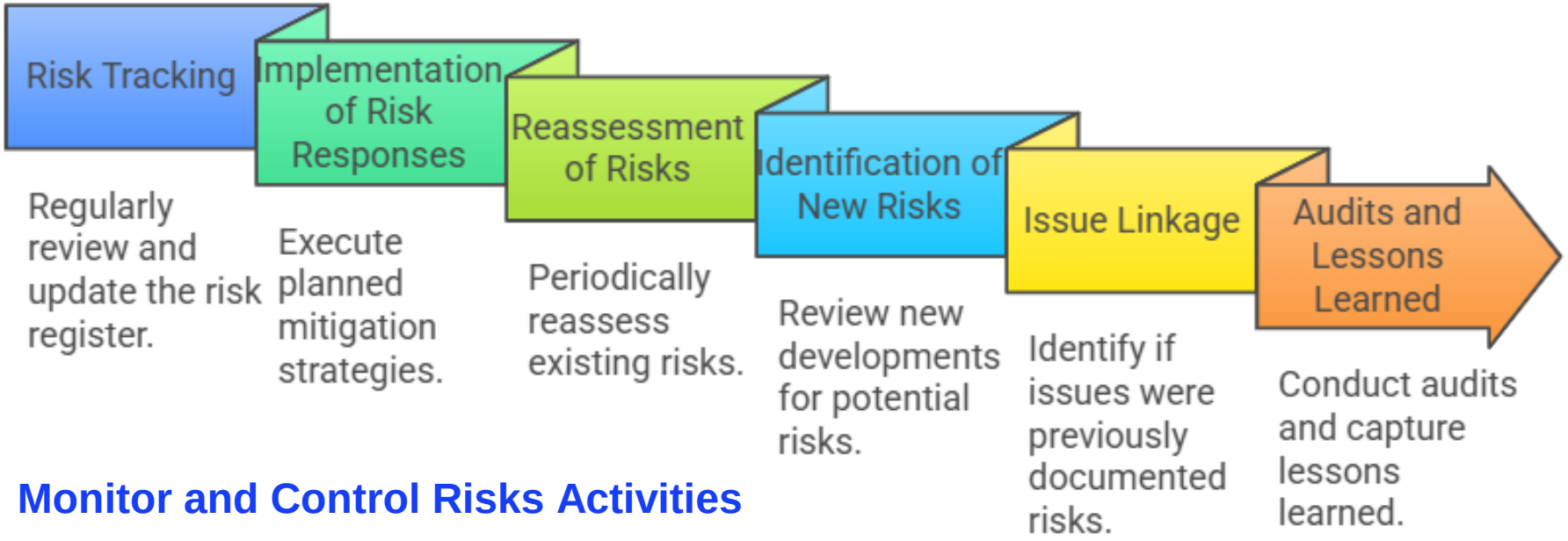
1. Identify the risk(s)
   - Failure to meet security standard and compromise quality
2. Choose an appropriate response strategy.
   - **Strategy:** Mitigation
   - **Action:** Contact vendor for update; explore alternatives
   - **Owner:** Project Manager, Development Team
   - **Time:** week 5-6

Project Risk Management Process

Monitor and Control Risks

- The ongoing process of tracking identified risks, detecting new ones, and ensuring that response plans are effective and adaptive
- It ensures
  - Implementing risk response plans
  - Evaluating risk process effectiveness throughout the project Tracking identified risks
  - Monitoring residual risks
  - Identifying new risks

Project Risk Management Process



Monitor and Control Risks Activities

Project Risk Management Process

Monitor and Control Risks - Integration with Other Project Processes

- Risk monitoring often overlaps with:
  - **Change control:** New risks may require changes in scope, schedule, or budget.
  - **Stakeholder communication:** Keep stakeholders informed of major risks.
  - **Quality management:** Quality issues often trigger risk reevaluations.
  - **Issue management:** Escalated risks become issues,
    - So coordination is essential.

Project Risk Management Process

Monitor and Control Risks - Example

- **Risk:** "If we use an external library that's poorly maintained, we might face compatibility issues later."
- **Monitoring:** The team tracks the release schedule of the library.
- **Control:** When the library misses two updates, the team escalates it, increases monitoring frequency, and starts prototyping an alternative.
- **Outcome:** The risk materializes (compatibility breaks), but the mitigation plan is ready, the team switches libraries with minimal disruption.

Project Risk Management

What is likely to happen in a project if identified risks are not resolved on time?

Unresolved project risks in a timely manner can create issues during project execution.

Project Issue Management

- An issue is a current problem that is impacting the project or could impact it if not resolved quickly.
- Issue management is the process of identifying, tracking, and resolving problems that occur during a software project.
- These problems are usually current
  - Affect the project’s ability to meet its objectives, scope, time, cost, or quality.

Project Issue Management

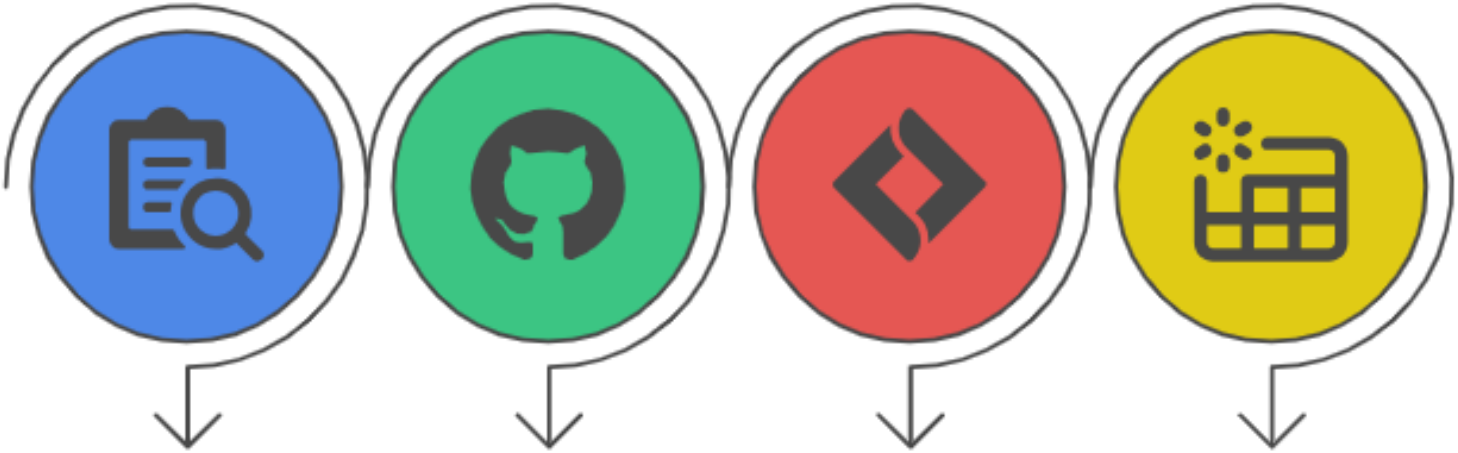
- **The types of issues can be**
  - Technical bugs
  - Communication breakdown
  - Stakeholder conflicts
  - Schedule delays
- **Examples:** A software bug affecting a key feature
  - A team member resigning mid-project
  - Integration problems between modules
  - Vendor delays

Project Issue Management

- **Steps of issue management:**
  - **Identification:** Logging and describing the issue
  - **Prioritization:** Assessing impact and urgency
  - **Assignment:** Assigning responsibility for resolution
  - **Tracking:** Monitoring issue progress
  - **Resolution:** Fixing the issue and verifying
  - **Closure:** Documenting the resolution and closing the issue

Project Issue Management

- **Tools for Issue Tracking**



Issue Log

Simple log for tracking issues

GitHub Issues

Issue tracking within GitHub repositories

Jira

Project management tool for issue tracking

Trello

Visual board for organizing tasks

Project Issue Management

- **Tools for Issue Tracking**
  - **Issue Log**
    - A document or system used to **track, manage, and monitor issues** that arise during the lifecycle of a project.
    - It includes
      - **Issue ID, description, identified date, reported by, assigned to, priority/severity, status, resolution date, resolution detail**

Project Issue Management

- **Tools for Issue Tracking**
  - Issue Log Example: **Online Learning Management System**

Issue ID	Description	Identified Date	Reported by	Priority	Assigned To
IS-01	LMS login not working			High	Dev Team

Status	Resolution Date	Resolution Detail
Open	-	

Project Issue Management

- **Risk Vs Issue**
  - **Risks Can Turn into Issues**
    - When a risk is not mitigated properly, it can materialize into a real issue. **Example**
      - **Risk:** “Third-party authentication API may be unreliable.”
      - **Issue:** During OLMS login, the API fails and students cannot log in (now an issue).
    - Every issue may be originated from an unmanaged risk

Project Issue Management

- **Risk Vs Issue**
  - **Risk Management Reduces Issues**
    - Effective risk identification and mitigation reduce the **number and severity of issues** during project execution.
  - **Issues Help Refine the Risk Register**
    - Issues encountered in a project provide **lessons learned** and help in identifying similar risks for future projects.
    - **Example:** LMS performance issues in the current semester highlight the risk of poor scalability in the next release.

Project Issue Management

- **Risk Vs Issue**
  - **Risk:** Future, uncertain
  - **Issue:** Present, actual
  - Risks become issues when realized.
  - Managing risks proactively prevents issues.
  - Managing issues reactively feeds into better risk identification.
  - Together, they form a **feedback loop** that strengthens project control.
  - Hence, risk register and issue log should be linked.

The Role of Project Manager

The project manager's role in risk and issue management:

- Ensure risks are being managed and tools are being used effectively.
- Ensure people respond to the risks and issues based on their expertise and knowledge.

Summary

- A well-structured risk response plan ensures that software projects stay on track despite uncertainties.
- By proactively identifying and managing risks, teams can avoid costly delays and failures.
- Risk response is not just about choosing a strategy, it’s about
  - Taking action
  - Assigning accountability, and
  - Keeping track.

Summary

- Every good risk response plan follows these steps.
  - Identify the risk clearly
  - Assess likelihood and impact
  - Select the response strategy
  - Define action steps
  - Assign responsibility
  - Set a timeline for actions

Summary

- Issue management addresses problems that have already occurred.
  - While risks are potential events, issues are current challenges.
- Understanding how to address issues promptly and effectively is a vital skill for any project manager.
- Effective issue management is a cornerstone of successful software project execution.

References

1. *PMI. (2013). A Guide to the Project Management Body of Knowledge (5th ed.). PMI, Inc.*
2. *Bob Hughes, M. C. (2011). Software Project Management (5th ed.). McGraw-Hill.*